A User-Centric Threat Model and Repository for Cyber Attacks

Prerit Datta[1], Sara Sartoli[1], Luis Felipe Gutierrez[1], Faranak Abri[1], Akbar Siami Namin[1], and Keith S. Jones[2]
[1] Department of Computer Science, [2] Department of Psychology, Texas Tech University
{prerit.datta,sara.sartoli,luis.gutierrez-espinoza,faranak.abri,akbar.namin,keith.s.jones}@ttu.edu

ABSTRACT

There are several threat-modeling schemes and most notably the one introduced by Microsoft, called STRIDE. These threat modeling schemes offer useful taxonomies that can be further utilized to capture key features and security aspects of software applications. However, to the best of our knowledge, the literature lacks a "user-centric" threat-modeling scheme through which the security features are perceived from a user's perspective. This paper introduces UC-STRIDE, a user-centric threat model, which enables identification of threats and communicates their associated risks from the user's point of view. UC-STRIDE extends the Microsoft STRIDE threat model to incorporate important aspects of threat modeling interested for the end users such as valuable assets, attack targets, and how the attack is launched. We introduce a repository of over 100 cyber attacks and their descriptions annotated with their immediate technical and non-technical consequences called "*CogSec*". The repository can be useful in communicating risks with stakeholders who may have little to no cybersecurity expertise. To illustrate the applicability of UC-STRIDE, we present a use case through which the application of the introduced user-centric threat modeling is demonstrated.

CCS CONCEPTS

• **Software and its engineering** → **Requirements analysis**; • **Security and privacy** → **Software and application security**;

KEYWORDS

Threat Modeling, Microsoft STRIDE, User-centric threat modeling

ACM Reference Format:
Prerit Datta[1], Sara Sartoli[1], Luis Felipe Gutierrez[1], Faranak Abri[1], Akbar Siami Namin[1], and, Keith S. Jones[2]. 2022. A User-Centric Threat Model and Repository for Cyber Attacks. In *The 37th ACM/SIGAPP Symposium on Applied Computing (SAC '22), April 25–29, 2022, Virtual Event, .* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3477314.3507315

1 INTRODUCTION

Threat modeling is a vital step in the software development lifecycle (SDLC). It is a process to identify security problems at the application level early on in the SDLC [6]. The process should be undertaken at the beginning of software development in order to reduce the cost of adding security features to the application at the later stages. Threat modeling has been used in software industry and more specifically in security for quite some time. The newer and more systematic threat modeling methodologies such as the Microsoft's "STRIDE" threat model tends to provide a holistic approach for modeling threats [4]. STRIDE is an acronym for **S**poofing, **T**ampering, **R**epudiation, **I**nformation Disclosure, **D**enial of Services, and **E**levation of Privileges. Although the Microsoft's STRIDE threat modeling scheme is effective and can be considered as the state-of-the-art of practice, it is designed mainly from an attacking point of view.

This paper introduces a user-centric view of STRIDE, called UC-STRIDE, an extension to the STRIDE model by introducing the concept of immediate technical and non-technical consequences to aid security designers, developers and stakeholders to better understand the attacks from not only the attacker's perspective but also its impact on users. The key motivation of designing a user-centric threat model is that it allows the system, software, and security designers to focus on potential vulnerabilities and misuses from the end-users perspective. The stakeholders may not be interested in technical aspects of cyber attacks in general and/or they may only care about the consequences of such attacks to them. Furthermore, the stakeholders may have different levels of technical concepts and may not comprehend the technical consequences of the cyber attacks the same. A user-centric threat model enables the designers and security experts to take into account threats and to liaise with the requirements for the desired system in a way that is comprehensible to everyone involved. This paper makes the following key contributions:

- A user-centric threat model based on STRIDE is introduced.
- A repository of attacks, called *CogSec*, along with their technical and non-technical consequences is built.
- A case study through which the feasibility and mechanic of the the introduced user-centric threat model is demonstrated.

2 UC-STRIDE: THE METHODOLOGY

A research team comprising of two faculty members specializing in Computer Science and Psychology (Human factors) and several research assistants majoring in Computer Science were involved in the brainstorming and creation of UC-STRIDE. The starting point of the work was the conventional STRIDE threat model where the research team collected an exhaustive list of attack descriptions and scenarios and classified them with respect to the original STRIDE taxonomy. The steps involved in the creation of UC-STRIDE model are briefly described below:

(1) A repository of security attacks and cyber threats, called "*CogSec*" (Cognitive Security) was created. The collection strategy was based on the STRIDE grouping of attacks: **S**poofing, **T**ampering, **R**epudiation, **I**nformation Disclosure, **D**enial of Services, and **E**levation of Privileges.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
SAC '22, April 25–29, 2022, Virtual Event,
© 2022 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-8713-2/22/04.
https://doi.org/10.1145/3477314.3507315

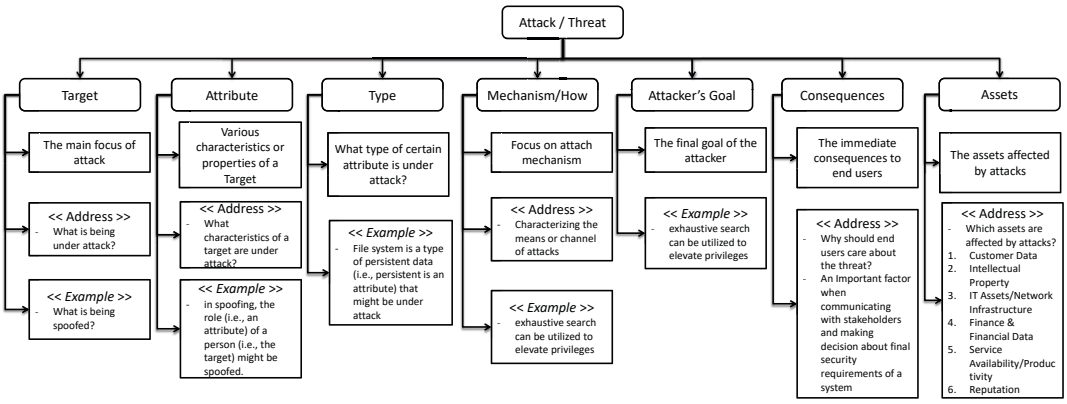

Figure 1: The user-centric features common to all threat types.

(2) For each security attack/threat, three distinct descriptions were collected. The research team searched online resources to find the best descriptions of each cyber threat. The intuition of collecting three distinct descriptions for each threat was due to the diversity of the perceived attacks.
(3) For each attack, an attack scenario for each description was analyzed and then a set of (common) features was built.
(4) For each attack, the immediate technical/non-technical consequences were identified.
   – *Technical.* For each attack scenario, the technical consequences of each attack to end-users were identified. The technical consequences were identified by the team members with Computer Science background.
   – *Non-Technical.* Similarly, for each attack scenario, the immediate and non-technical consequences of each attack to end-users were identified. The non-technical consequences immediate to end-users were identified by the team member with Psychology and Human-factors research background.
(5) The set of common features were then used as a starting point to create a concept map (or a mind map) for each category of STRIDE attack classification (e.g., Spoofing).
(6) The concept map and the features were refined and analyzed throughout several brainstorming sessions.

3 *CogSec*: THE FEATURES

The research team searched and collected various forms of attacks or threats along with their descriptions corresponding to each category (i.e., S, T, R, I, D and E) that together constitute the STRIDE model. To have a comprehensive list of attacks or threats, various technical blogs (such as SANS, Symantec), MITRE Vulnerability database (CVE), as well as technical conferences and journals were chosen to create the repository. Table 1 reports the number of attacks and their descriptions that were analyzed to create the repository. To comprehend the meaning and semantics of each attack, the authors searched and collected three different descriptions of each attack[1].

The research team utilized various descriptions (three versions) of each cyber threat and came up with the technical and non-technical consequences of each cyber attack that are *immediate* to end-users. The *immediate* consequences are those that affect the users directly and they are observable immediately; whereas, *non-immediate* consequences are those that affect the users in long range. For instance, a cyber attack may cause an end-user credit card details to be stolen; whereas, the long-term consequence of credit card theft could be identity theft or fraud, the immediate consequence to the end-user can be described as financial loss.

Table 1: # attacks and descriptions studied for UC-STRIDE.

	# Concrete Attacks	# Attack Descriptions
Spoofing	12	36
Tampering	15	45
Repudiation	9	27
Information Disclosure	19	57
Denial of Services	25	75
Elevation of Privileges	22	66
Total	102	306

Figure 1 enlists seven common features considered for the taxonomy. The seven features focus on aspects of cyber threats that are important to end users. The features are listed as: 1) "*Target*" of the attack, 2) "*Attribute*" of the attack, 3) "*Type*" of the attack, 4) "*Mechanism*" describing how the attack is launched, 5) "*Goal*" the attacker's goal, 6) "*Consequences*" to users, and 7) "*Assets*" targeted.

4 THE UC-STRIDE APPLICATION PROCESS

The process is formulated based on the methodology presented by Microsoft for using the original STRIDE model [1]. The process consists of four steps as follows:

Step 1. Characterizing the System through Architectural Model. System and in particular security designers need to identify the software components and the relationship between them in the underlying system. Hence, the first step involves constructing an architectural model of the system under design. Conventional STRIDE methodology suggests building a Data-Flow Diagram (DFD) [1]. The diagram will be used later in the process to identify associated threats. Other architectural models such as activity diagram or block diagrams can also be integrated in the threat modeling methodology [3, 5]. During this step, the security designer must characterize security assumptions (e.g., the presence of access control or authentication mechanism).

Step 2. Apply STRIDE to Elements in the Architectural Model. Once the system under design is characterized and an architectural model is built, the conventional STRIDE threat modeling is applied to the elements in the model of the system (e.g. DFD). Each element type

[1]Repository: https://github.com/asiamina/UC-STRIDE.CogSec.Model.Repository

Table 2: Data-flow Diagram to STRIDE mapping.

	Category					
(I) Dataflow Diagram Elements	S	T	R	I	D	E
External Entity	√		√			
Data Flow		√		√	√	
Data Store		√		√	√	
Processing Node	√	√	√	√	√	√
(II) Block Diagram Elements	S	T	R	I	D	E
Human Agents	√		√			
Agents	√	√	√	√	√	√
Channels		√		√	√	
Storage		√		√	√	
Access	√	√	√	√	√	√
Components	√	√	√	√	√	√
(III) Activity Diagram Elements	S	T	R	I	D	E
Actor	√		√			
Activity Edge		√		√	√	
Condition		√	√	√	√	
Activity	√	√	√	√	√	√

of the model can be mapped to at least one threat category. Table 2 shows an example of such a mapping.

Step 3. Elicit Possible Threats and the Consequences of each Threat. For each mapping between a threat category and an element in one of the diagrams, we built an abstraction graph of concrete threats that need to be considered. As an example, the graph for spoofing is shown in Figure 3. The abstraction graph has a tree-structure containing seven dimensions of each threat category and is annotated with immediate consequences to end-users' and the assets targeted. The presented hierarchical structure provides an easy to navigate graphical tool and also helps to better understand each of threat categories. More importantly, it provides the security designer with the user consequences of threats which can be useful to communicate with the stakeholders at the design time. For a complete set of abstraction graphs (i.e., concept maps), please visit the GitHub repository of this research.

Step 4. Specifying Security Requirements. In this stage, the security designer should elicit and document security requirements. The security threats elicited and assessed in Stage 3 are used as a basis of security requirements elicitation process [3]. Security designers need to find a balance between security and usability goals. Therefore, in this step, it is important to know which threats are realistic and which threats users care about. We leave risk assessment out and assume that security designer has prepared a list of prioritized threats based on results of risk assessment.

5 *CogSec* IN PRACTICE: A USE CASE

This section elucidates the usefulness and the application of UC-STRIDE model with the help of a use-case example. The use-case described in this section has been adapted from [2]. Pet shop 4.0 is a web-based application developed using Microsoft's .Net framework 2.0 and ASP.Net. The application was selected for its simplicity and to illustrate the application of UC-STRIDE in practice. However, UC-STRIDE model can be applied to application of any size. Due to space limitations, we only list some of the DFD elements in Figure 2 containing the level 0 DFD (Data-Flow Diagram) of the Pet Shop application.

Table 3 details how the various DFD data store can be mapped to immediate technical and non-technical consequences based on (non)-technical consequences listed in Table 4 for each classes of

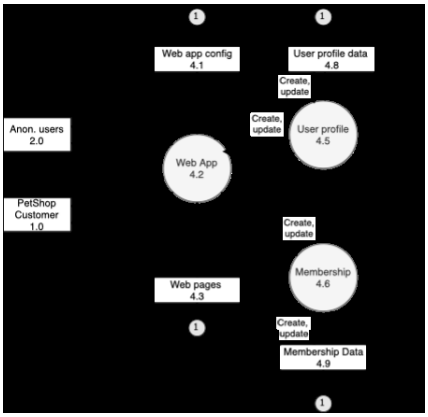

Figure 2: Level 0 DFD for Pet Shop shop application (adapted from [2]).

Table 3: Mapping PetShop DFD elements to UC-STRIDE Consequences.

Description	Consequences					
	S	T	R	I	D	E
DFD Element Type: Data Stores.						
- Web application configuration (4.1)		T2, T8		I3, I4		
- Web pages (4.3)		T11		I3, I10		
- User profile data (4.8)		T6, T12		I11		
- Membership data (4.9)		T6, T12		I11		

S:Spoofing, T: Tampering, R:Repudiation, I: Info. Disclosure, D: Denial-of-service, E: Elevation of privilege. The number corresponds to the consequences in Table 4.

attacks. For the complete DFD elements to consequences mapping, please refer to the GitHub repository of the project.

For Pet Store web application, the *data stores* and *data flows* are potential sources of attacks. Web application configuration data (4.1) can be exploited by the attacker using cross-site scripting attacks **(T2)** and buffer overflow **(T8)** attacks. The Web pages (4.3) can be defaced **(T11)** by the attacker to ruin the reputation of the website.

Additionally, abstraction graphs, such as defined Figure 3, can provide useful aide in describing the consequences and mapping each threat to dimensions. For example, an external entity Pet Shop Customer (1.0) **(target)** ⟶ ID **(attribute)** can be spoofed by an attacker ⟶ by Forging ID **(how)** for reasons such as stealing credentials, gaining (client) information, damaging the (client's) reputation **(attacker goals)** ⟶ which can lead to **(consequences)** and target ⟶ customer data, customer's reputation **(assets)**.

6 CONCLUSION AND FUTURE WORK

There are a number of taxonomies designed specifically for classification of cyber security attacks. These taxonomies are designed for addressing the needs of technical developers and system engineers. However, from end-user's point of view, as one of the major stakeholders, these taxonomies are less useful or informative. Users can contribute to the design of systems if they understand the meaning and more importantly the consequences of every attack and as a result they can help in mitigating the risks.

Table 4: The "partial" immediate technical/non-technical consequences of Tampering and Information Disclosure attacks.

		Immediate Consequences	
		Technical	**Non-Technical**
...			
(T) Tampering Attacks			
2	Cross-site scripting (XSS)	Malicious script executed by Web application	1) The cyber-attacker performed actions on an Internet site as if she was you. 2) The cyber-attacker changed the appearance of an Internet site. 3) The cyber-attacker accessed your information stored in an Internet site.
6	Tampering with acc. Parameter in URL	Break into user's account	
8	Buffer overflow	Corrupting execution logic of Web application	1) The cyber-attacker made your computer crash., 2) The cyber-attacker made your computer run software that your computer did not intend to run.
12	Database tampering	Manipulation of user data	The cyber-attacker modified your information within an Internet database.
(I) Information Disclosure Attacks			
3	Through error message (exception)	Exposing logical relationships of the system	
4	Buffer overflow (code injection)	Corrupting execution logic of a Web application	1) The cyber-attacker made your computer crash., 2) The cyber-attacker made your computer run software that your computer did not intend to run.
10	Inadequate file protection	Exposure of system/user data	
11	Inadequate database security	Exposure of system/user data	
...			

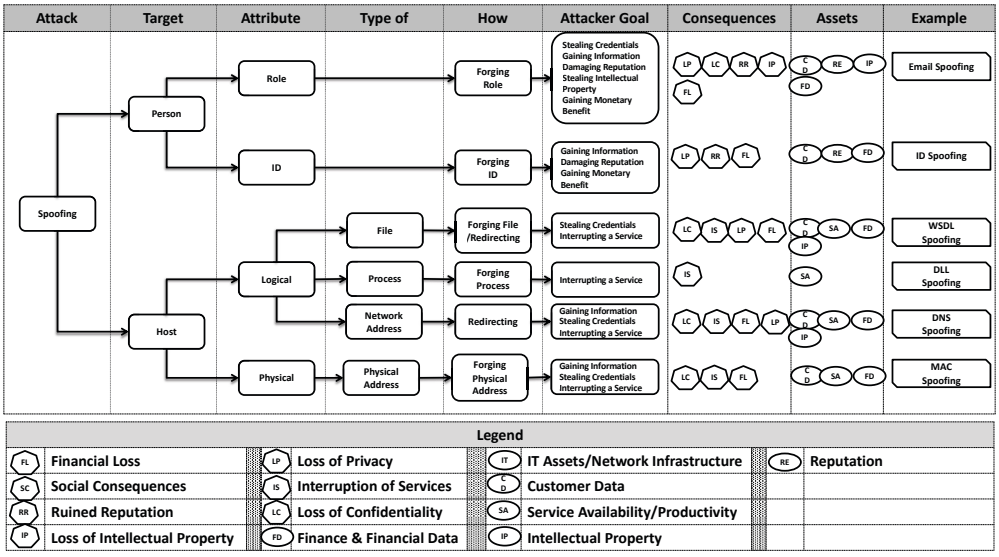

Figure 3: The concept map created for Spoofing.

This paper introduced a user-centric classification of cyber attacks, called UC-STRIDE. The introduced taxonomy is built on top of the Microsoft's STRIDE threat model, and thus embraces major security threats related to Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, and Elevation of privilege. In creation of the UC-STRIDE model, several factors and dimensions are taken into account including 1) the Target of the attack, 2) the Attribute of the attack, 3) The Type of the attack, 4) the Goal of the attack, 5) the way (i.e., How) the attack is launched, and 6) the Consequences to the end users. This taxonomy allows security designers to have end-users in mind when making decisions about security features. It also facilitates the communication between technical and non-technical stakeholders.

As a future research direction and enhancement to UC-STRIDE, the proposed taxonomy can be enhanced if information regarding the impacts and risks of each attacks are also integrated and presented to the end users. The conventional definition of risks is usually formulated in terms of likelihood and the impact of each attack. A proper quantification of risks can be beneficial to prioritize each attack, and thus plan a better risk mitigation strategy.

ACKNOWLEDGEMENT

This research work is supported by National Science Foundation (NSF) under Grant No: 1564293.

REFERENCES

[1] Howard and Lipner. 2006. *The security development lifecycle.* Redmond: Microsoft Press.
[2] M. Howard and S. Lipner. 2006. The Security Development Lifecycle. Microsoft Press.
[3] Johnstone. 2010. Threat modeling with Stride and UML. In *Proceedings of the 8th Australian Information Security Management Conference.*
[4] OWASP. 2018. Application Threat Modeling. https://www.owasp.org/index.php/Application_Threat_Modeling
[5] Schaad and Borozdin. 2012. TAM 2: automated threat analysis. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing.*
[6] Adam Shostack. 2014. *Threat Modeling: Designing for Security* (1st ed.). Wiley Publishing.